Efficient Revocable Identity-Based SM9 Signature Algorithm Postprint

Authors: Zhang Boxin, Geng Shengling, Qin Baodong

Date: 2022-05-10T11:22:57Z

Abstract

SM9-IBS is an identity-based signature algorithm industry standard published by China in 2016. Although identity-based signature algorithms reduce the complexity of system management of user public keys, they suffer from the difficult problem of key revocation. Furthermore, the special structure of SM9 makes existing techniques not fully applicable. To this end, a revocable SM9 identity-based signature algorithm is proposed, enabling rapid revocation and update operations of user signing privileges. The algorithm introduces a complete subtree, with which the key center generates temporary signing keys for each legitimate user; only signatures generated using this key can be successfully verified. In terms of security, the algorithm is proven to satisfy existential unforgeability under adaptive chosen-message and identity attacks in the random oracle model. In terms of efficiency, when the number of system users is large and the number of revoked users is small during the key update phase, the time overhead for the key center to update user signing keys is far less than the update technique of Boneh et al.

Full Text

Preamble

Vol. 39 No. 9
Application Research of Computers
ChinaXiv Partner Journal

Efficient Revocable SM9 Identity-Based Signature Algorithm

Zhang Boxin[1], Geng Shengling[2], Qin Baodong[1]†
(1. School of Cyberspace Security, Xi' an University of Posts & Telecommunications, Xi' an 710121, China;
2. School of Computer, Qinghai Normal University, Xining 810008, China)

Abstract: SM9-IBS is an industry standard for identity-based signature algorithms issued by China in 2016. Although identity-based signature algorithms reduce the complexity of public key management, they face the challenge of key revocation. Moreover, SM9' s special algebraic structure makes existing revocation techniques inapplicable. This paper proposes an efficient revocable SM9 identity-based signature algorithm that enables rapid revocation and updating of user signing privileges. The algorithm introduces a complete subtree through which the Key Generation Center (KGC) generates temporary signing keys for each legitimate user, ensuring only signatures produced with these keys pass verification. Security-wise, the algorithm is proven existentially unforgeable under adaptive chosen message and identity attacks in the random oracle model. In terms of efficiency, when the system has many users but few revoked ones, the KGC' s time overhead for updating user signing keys is significantly lower than Boneh et al.' s update technique.

Keywords: identity-based cryptosystem; SM9 signature algorithm; key revocation; complete subtree

---

1.1 Symbol Description

In this paper, if S is an algorithm, then $s \leftarrow S$ denotes executing algorithm S with output s; if S is a set, then $s \leftarrow S$ denotes uniformly random selection of an element s from set S. $A||B$ represents concatenation of two bit strings A and B.

In the SM9 algorithm:
a) $H_v(\cdot)$ is a cryptographic hash function that takes arbitrary bit strings as input and outputs v bits.
b) $H_2(\cdot)$ is a cryptographic function that takes as input a cryptographic hash function $H_v(\cdot)$, a bit string Z, and an integer p, and outputs an integer in $\mathbb{Z}_p^*$.

1.2 Bilinear Mapping

Let $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ be cyclic groups of prime order p, with g_1 and g_2 as generators of $\mathbb{G}_1$ and $\mathbb{G}_2$ respectively. A bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ must satisfy the following properties:

a) **Bilinearity:** For any $\alpha, \beta \in \mathbb{Z}_p^*$, $e(g_1^\alpha, g_2^\beta) = e(g_1, g_2)^{\alpha\beta}$.

b) **Non-degeneracy:** $e(g_1, g_2) \neq 1_{\mathbb{G}_T}$.

c) **Computability:** For any $h_1 \in \mathbb{G}_1$ and $h_2 \in \mathbb{G}_2$, there exists an efficient polynomial-time algorithm to compute $e(h_1, h_2)$.

1.3 Hardness Assumption

This section introduces the q-Strong Diffie-Hellman (q-SDH) problem, upon which the SM9 signature algorithm' s security against existential forgery under adaptive chosen message and identity attacks (EU-CMIA) relies, as proposed in [?]. Let $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ be a bilinear map, with g_1 and g_2 as generators of $\mathbb{G}_1$ and $\mathbb{G}_2$ respectively, where groups $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ have prime order p.

The q-SDH problem in bilinear groups is defined as follows:

Definition 1 (q-SDH Problem). Given $q+1$ group elements $(g_1, g_1^a, g_1^{a^2}, \dots, g_1^{a^q}, g_2, g_2^a)$ where $a \in \mathbb{Z}_p^*$ is unknown, find a pair $(c, g_1^{1/(a+c)})$ with $c \in \mathbb{Z}_p^*$. If the probability of solving the q-SDH problem in polynomial time is negligible, the q-SDH assumption holds.

1.4 SM9-IBS Identity-Based Signature Algorithm

The SM9 identity-based signature algorithm (SM9-IBS) consists of four (probabilistic) polynomial-time algorithms: system parameter generation, user key extraction, signature generation, and verification. We briefly review the SM9-IBS algorithm below.

a) System Setup: The KGC first runs the algorithm SM9.Setup(1^λ) on input security parameter λ to generate a bilinear group $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2)$. Then it selects a random $s \in \mathbb{Z}_p^*$ as the master secret key. The system master public key and master secret key are $mpk = (g_1, g_2, g_2^s, H_2(\cdot))$ and $msk = s$ respectively. The system master public key serves as the default input for other algorithms (omitted for brevity).

b) User Key Extraction: For a user with identity id, the KGC computes the signing private key as $ds_{id} = g_1^{s \cdot H_2(H_v(id))}$, where $H_2(H_v(id)) \in \mathbb{Z}_p^*$. The signer' s public key is id and private key is ds_{id}.

c) Signing: A signer with identity id signs message M as follows:
1. Select random $r \in \mathbb{Z}_p^*$ and compute $w = g_2^r$.
2. Compute $h = H_2(H_v(M||id), w)$.
3. If $h = 0 \pmod{p}$, return to step 1.
4. Compute $l = (r - h) \cdot s^{-1} \pmod{p}$.
5. Compute $S = ds_{id}^l$.

The signature on M is $\sigma = (h, S)$.

d) Verification: Upon receiving signer's identity id, message M, and signature $\sigma = (h, S)$, the verifier:
1. Computes $w' = g_2^h \cdot e(S, g_2^{s \cdot H_2(H_v(id))})$.
2. Computes $h' = H_2(H_v(M||id), w')$.
3. Accepts the signature if $h' = h$; otherwise rejects.

Note 1: To improve signing and verification efficiency, the fixed value $e(g_1, g_2)$ computed in the original SM9-IBS algorithm can be included in the system public key, reducing one bilinear pairing operation in both signing and verification while adding one $\mathbb{G}_T$ element to the system public key.

The EU-CMIA security model for identity-based signature algorithms is similar to traditional signature schemes, except that besides trivial queries, the adversary can adaptively choose any identity to obtain its signing key and any message to obtain its signature. See [?] for details.

Under this model, we have:

Theorem 1 (Security of SM9-IBS [?]). In the random oracle model, if the q-SDH problem is hard, then SM9-IBS satisfies EU-CMIA security.

2.1 RIBS Definition

Figure 1 illustrates the system model of revocable identity-based signature (RIBS), which involves four entities:

a) Key Generation Center (KGC): Generates system master public key mpk and master secret key msk, creates long-term signing keys ds_{id} for users with identity id, generates update key sets uk_t, maintains user revocation list RL, and manages a full binary tree $Tree$ corresponding to all user identities.

b) Data Signer (DS): Computes digital signature σ on original message M.

c) Storage Server (SS): Stores user signatures.

d) Signature Verifier (SV): Verifies the validity of received signatures σ.

Definition 2 (RIBS Definition). A RIBS algorithm consists of eight (probabilistic) polynomial-time algorithms: system parameter generation, user registration, update node generation for non-revoked users, update key generation, temporary signing key generation, signing, verification, and user revocation.

a) Setup(1^λ): Executed by the KGC. Takes security parameter λ as input and outputs system master public key mpk and master secret key msk. The KGC maintains an initially empty revocation list RL and a full binary tree $Tree$ corresponding to all user identities.

b) Register(msk, id): Executed by the KGC. Takes master secret key and user identity id as input, adds id to $Tree$, and outputs the user' s long-term signing key ds_{id}.

c) KUNode$(Tree, RL, t)$: Executed by the KGC. Takes identity tree $Tree$, revocation list RL, and time t as input, and outputs the set of nodes $KUNodes$ for which update keys need to be generated.

d) UpdateK($msk, t, KUNodes$): Executed by the KGC. Takes master secret key msk, time t, and non-revoked node set $KUNodes$ as input, and outputs update key set uk_t, broadcast to system users via public channel.

e) TempK(ds_{id}, uk_t): Executed by the data signer. Takes user' s long-term signing key ds_{id} and update key set uk_t as input. If the user is not revoked, it outputs a temporary signing key $ts_{id,t}$ (containing user identity id, update time t, and corresponding update node information θ); if revoked, it outputs $\perp$.

f) Sign($M, ts_{id,t}$): Executed by the data signer. Takes message M and user temporary signing key $ts_{id,t}$ as input, computes signature σ on M (containing time period and update node information), and sends (M, σ) to the storage server.

g) Verify(id, M, σ): Executed by the signature verifier. Takes signer' s identity id, message M, and signature σ as input, and outputs verification result (1 for valid, 0 for invalid).

h) Revoke(id, t, RL): Executed by the KGC. Takes revoked user identity id, time period t, and revocation list RL as input, adds (id, t) to RL, and returns the updated list.

2.2 Security Model

To characterize the key revocation mechanism, we describe existential unforgeability under adaptive chosen message and identity attacks (EU-CMIA) through a game between adversary and challenger. In this model, the adversary can query any user' s long-term signing key. When a user' s key is compromised (queried by the adversary) and revoked by the KGC at time t, the scheme' s security ensures the adversary cannot generate a valid signature for that user at time t.

Definition 3 (EU-CMIA Security of RIBS). Let $\mathcal{A}$ be any probabilistic polynomial-time adversary and $\mathcal{C}$ be a challenger. The security experiment $\mathrm{Exp}^{\text{EU-CMIA}}_{\text{RIBS},\mathcal{A}}(1^\lambda)$ is defined as:

a) Setup: Challenger runs Setup(1^λ) to generate master public key mpk and master secret key msk, initializes empty key revocation list RL and full binary tree $Tree$, and sends mpk to $\mathcal{A}$.

b) Queries: $\mathcal{A}$ can adaptively make polynomially many queries:

- **Long-term signing key query:** When $\mathcal{A}$ queries user id' s long-term signing key, challenger runs Register(msk, id) to generate key ds_{id} and returns it, adding id to set L_1.
- **Update key query:** When $\mathcal{A}$ queries update key for time t, challenger first runs KUNode$(Tree, RL, t)$ to get update node set $KUNodes_t$, then

runs UpdateK($msk, t, KUNodes_t$) to obtain update key set uk_t, and returns it.

- **Signature query:** When $\mathcal{A}$ queries signature on message M for identity id at time t, challenger first checks if id was revoked at or before t (assuming $\mathcal{A}$ always queries update key for time t first). If not revoked, challenger uses ds_{id} and uk_t to generate temporary signing key $ts_{id,t}$, then signs M to get σ and returns it, adding (id, t, M) to set L_2.
- **Revocation query:** When $\mathcal{A}$ queries revocation of identity id at time t, challenger adds (id, t) to revocation list RL.

c) Forgery: $\mathcal{A}$ outputs a forged signature (M^*, σ^*) for challenge identity id^* and time t^*, satisfying: - If id^* was not revoked at or before t^*, then $id^* \notin L_1$. - If id^* was revoked at or before t^*, then $(id^*, t, M^*) \notin L_2$ for any $t \leq t^*$.

d) Output: If the forged signature passes verification, $\mathcal{A}$ wins and challenger returns 1; otherwise returns 0.

$\mathcal{A}$' s success probability is $\Pr[\mathrm{Exp}_{\mathrm{RIBS},\mathcal{A}}^{\mathrm{EU\text{-}CMIA}}(1^\lambda) = 1]$. The RIBS algorithm is EU-CMIA secure if this probability is negligible in λ.

2.3 Algorithm Design

This section presents a revocable SM9 identity-based signature algorithm using complete subtree coverage (denoted CS-SM9-RIBS).

a) Setup(1^λ): On input security parameter λ: 1. KGC runs bilinear group generation to get $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, g_1, g_2)$. Assume each user identity is n bits long. KGC initializes a full binary tree $Tree$ of height $n+1$ and empty revocation list RL. 2. KGC selects random $s \in \mathbb{Z}_p^*$ and computes $P_{pub} = g_2^s$. The master public key is $mpk = (g_1, g_2, P_{pub}, H_2(\cdot))$ and master secret key is $msk = s$.

b) Register(msk, id): For user registration request with identity id: 1. KGC uses SM9 key extraction to generate signing key: $ds_{id} = g_1^{s \cdot H_2(H_v(id))}$. 2. KGC assigns a leaf node matching id, adds id to $Tree$, and sends ds_{id} to the user.

c) KUNode($Tree, RL, t$): On input identity tree $Tree$, revocation list RL, and time t: 1. Let X and Y be empty sets. For each non-leaf node θ in $Tree$, let θ_l and θ_r be its left and right children. Let $Path(id)$ be the set of nodes on the path from root to leaf node for identity id. 2. For each $(id, t') \in RL$ with $t' \leq t$, add $Path(id)$ to X. 3. For each node $\theta \in X$: if $\theta_l \notin X$, add θ_l to Y; if $\theta_r \notin X$, add θ_r to Y. 4. If $Y = \emptyset$, set $Y = \{\text{root}\}$. 5. Output $KUNodes = Y$.

Figures 2 and 3 show examples of update node selection. In these examples, the binary tree has height 4 (identity length = 3 bits). Blue nodes represent current update nodes; dashed nodes represent revoked users. Figure 2 shows no revocation ($KUNodes = \{\text{root}\}$). Figure 3 shows user with identity "3" revoked, yielding $KUNodes = \{00, 010, 1\}$.

d) UpdateK($msk, t, KUNodes$): On input master secret key msk, time t, and update node set $KUNodes$: 1. KGC computes update identity set $\{UID||t|\theta\}_{\theta\in KUNodes}$ where $UID||t|\theta$ concatenates time and node information. 2. For each identity in the set, KGC uses SM9 key extraction to generate key $ds_{UID||t|\theta}$. 3. The update key set is $uk_t = \{ds_{UID||t|\theta}\}_{\theta\in KUNodes}$, broadcast via public channel.

e) TempK(ds_{id}, uk_t): On input user' s long-term signing key ds_{id} and update key set uk_t: 1. If user is not revoked, there exists a unique common node $\theta \in Path(id) \cap KUNodes$. User finds corresponding update key $ds_{UID||t|\theta}$ and defines temporary signing key as $ts_{id,t} = (ds_{id}, ds_{UID||t|\theta}, \theta)$. If revoked, output $\perp$. 2. Return $ts_{id,t}$.

f) Sign($M, ts_{id,t}$): On input message M and temporary signing key $ts_{id,t}$: 1. If $ts_{id,t} = \perp$, abort. 2. Recover ds_{id} and $ds_{UID||t|\theta}$ from $ts_{id,t}$, let $M' = M||t||\theta$. 3. Generate first signature component: $\sigma_1 \leftarrow$ SM9.Sign(ds_{id}, M'). 4. Generate second signature component: $\sigma_2 \leftarrow$ SM9.Sign($ds_{UID||t|\theta}, M'$). 5. Output final signature $\sigma = (\sigma_1, \sigma_2, \theta)$.

g) Verify(id, M, σ): On input signer' s identity id, message M, and signature $\sigma = (\sigma_1, \sigma_2, \theta)$: 1. Check if node θ is on $Path(id)$. If not, return 0. 2. Compute $M' = M||t||\theta$ and verify σ_1 using SM9.Verify with identity id: $b_1 \leftarrow$ SM9.Verify(id, M', σ_1). If $b_1 = 0$, return 0. 3. Verify σ_2 using SM9.Verify with update identity $UID||t|\theta$: $b_2 \leftarrow$ SM9.Verify($UID||t|\theta, M', \sigma_2$). If $b_2 = 0$, return 0. 4. Return 1 if all checks pass.

h) Revoke(id, t, RL): On input revoked user identity id, time period t, and revocation list RL: 1. If $(id, t') \in RL$ for some $t' \leq t$, return RL. 2. Otherwise, add (id, t) to RL and return the updated list.

Correctness: For signature $\sigma = (\sigma_1, \sigma_2, \theta)$ on message M generated by Sign($M, ts_{id,t}$), σ_1 is a valid SM9 signature under identity id, so step (b) of verification succeeds. If user id is not revoked at time t, by the update node selection algorithm, there exists a common node $\theta \in Path(id) \cap KUNodes$, so step (a) passes. Since σ_2 is a valid SM9 signature under identity $UID||t|\theta$, step (c) succeeds. Thus, CS-SM9-RIBS satisfies correctness.

---

3 Security Analysis

Theorem 2 (Security of CS-SM9-RIBS). Let $\mathcal{A}$ be any probabilistic polynomial-time adversary attacking the EU-CMIA security of CS-SM9-RIBS with success probability ϵ. Then there exists another probabilistic polynomial-time algorithm $\mathcal{B}$ that can forge an SM9-IBS signature with the same probability ϵ.

Proof. We construct a simulator $\mathcal{B}$ that uses $\mathcal{A}$ as a subroutine to forge an SM9-IBS signature in the EU-CMIA model. $\mathcal{B}$ simulates $\mathcal{A}$' s environment as follows:

a) Setup: $\mathcal{B}$ queries the SM9-IBS challenger to obtain challenge master public key $mpk = (g_1, g_2, P_{pub}, H_2(\cdot))$ and sends mpk to $\mathcal{A}$. $\mathcal{B}$ initializes empty revocation list RL, full binary tree $Tree$, and empty sets L_1 and L_2.

b) Queries: $\mathcal{B}$ answers $\mathcal{A}$' s queries as:

- **Long-term signing key query:** For query on identity id, $\mathcal{B}$ forwards id to the SM9-IBS challenger to obtain ds_{id} and returns it, adding id to L_1.
- **Update key query:** For time t, $\mathcal{B}$ computes $KUNodes_t$ using KUNode($Tree, RL, t$). For each $\theta \in KUNodes_t$, $\mathcal{B}$ queries the challenger for key $ds_{UID||t|\theta}$ and returns the set $uk_t = \{ds_{UID||t|\theta}\}_{\theta \in KUNodes_t}$ to $\mathcal{A}$, adding all $UID||t|\theta$ to L_2.
- **Signature query:** For query (id, t, M), $\mathcal{B}$ first checks if id was revoked at or before t (assuming $\mathcal{A}$ queries update keys first). If revoked, return $\perp$. Otherwise, $\mathcal{B}$ uses $ds_{UID||t|\theta}$ to generate σ_2. For σ_1, if $id \in L_1$, use ds_{id}; otherwise query the challenger for signature on $M||t||\theta$ under identity id. Return $\sigma = (\sigma_1, \sigma_2, \theta)$.
- **Revocation query:** For revocation of id at time t, $\mathcal{B}$ adds (id, t) to RL.

c) Forgery: $\mathcal{A}$ outputs forged signature (M^*, σ^*) for challenge identity id^* and time t^*.

d) Output: If σ^* is a valid CS-SM9-RIBS signature, then: - If $id^* \notin L_1$, $\mathcal{B}$ outputs $(M^*||t^*||\theta^*, \sigma_1^*)$ as a forgery for SM9-IBS. - If $id^* \in L_1$, then id^* must have been revoked before t^*. By the update node generation algorithm, $KUNodes_{t^*}$ contains no nodes on $Path(id^*)$, so $\mathcal{B}$ never queried the challenger for $ds_{UID||t^*||\theta^*}$. Thus $\mathcal{B}$ can output $(M^*||t^*||\theta^*, \sigma_2^*)$ as a forgery.

$\mathcal{B}$ perfectly simulates $\mathcal{A}$' s environment. If $\mathcal{A}$ successfully forges a CS-SM9-RIBS signature, $\mathcal{B}$ forges an SM9-IBS signature with the same probability.

---

4 Performance Analysis

Currently, besides directly applying Boneh-Franklin' s generic revocation approach to SM9-IBS, no published revocation algorithms exist for SM9-IBS. However, excellent revocation algorithms exist for other identity-based signatures [?, ?]. This section compares performance theoretically and experimentally among: (1) our CS-SM9-RIBS, (2) original SM9-IBS [?], and (3) BF-SM9-RIBS (Boneh-Franklin approach).

Table 1 summarizes theoretical performance comparisons. We use symmetric bilinear pairings where group order is prime p. N and R denote total users and revoked users; "E" and "P" denote group exponentiation and pairing operations (ignoring other operations). “-” indicates non-existence; “0” indicates negligible cost.

In BF-SM9-RIBS, users store only per-period temporary keys generated by the KGC. The update strategy concatenates identity id with time period t as $id||t$ to generate temporary signing keys sent via secure channels, yielding $O(N-R)$ update complexity that grows linearly with user count. Our CS-SM9-RIBS uses complete subtree technology, requiring only $O(\log(N/R))$ update keys on average, transmittable over public channels.

Table 1: Performance Comparison

Metric	SM9-IBS	BF-SM9-RIBS	CS-SM9-RIBS
System public key size	$2\|\|p\|\|$	$2\|\|p\|\|$	$2\|\|p\|\|$
System private key size	$\|\|p\|\|$	$\|\|p\|\|$	$\|\|p\|\|$
Long-term key size	-	0	$\|\|p\|\|$
Update key size	-	$(N-R)\|\|p\|\|$	$(\log(N/R))\|\|p\|\|$
Temporary key size	$\|\|p\|\|$	$\|\|p\|\|$	$2\|\|p\|\|$
Long-term key time	-	0	1E
Update key time	-	$(N-R)$E	$(\log(N/R))$E
Temporary key time	-	0	0
User revocation time	-	0	0
Signing time	$1P+2E$	$1P+2E$	$2P+4E$
Verification time	$1P+2E$	$1P+2E$	$2P+4E$
Key update channel	-	Secure	Public
Revocation support	No	Yes	Yes

Both BF-SM9-RIBS and CS-SM9-RIBS support revocation while preserving SM9-IBS parameters. BF-SM9-RIBS requires no long-term key storage and has single-element temporary keys, but suffers high update complexity and needs secure channels. CS-SM9-RIBS stores one long-term key; its temporary key contains two group elements but includes the long-term key without extra storage. Signature size, signing time, and verification time are double that of SM9-IBS due to the additional signature operation for revocation.

Experimental Setup: We implemented simulations on a 2.4GHz Intel i5-9300H CPU with 16.0GB RAM, Windows 10, using the JPBC library with Type-F elliptic curves. **Table 2** shows average execution times (excluding negligible operations and one-time setup). For update key generation with $N = 100$ users and no revocations, BF-SM9-RIBS generates 100 keys while CS-SM9-RIBS generates only 1.

Table 2: Average Execution Time Comparison

Algorithm	User Registration	Update Key Gen.	Temp Key Gen.	Signing	Verification
SM9-IBS [?]	0.006s	-	-	0.043s	0.478s
BF-SM9-RIBS [?]	0.006s	0.685s	0.042s	0.006s	0.467s
CS-SM9-RIBS (Sec 2.3)	0.006s	0.006s	0.088s	0.852s	0.478s

For deeper comparison, we measured update key generation with N up to 8192 and R varying from 0 to 100 (random positions). **Figure 4** shows CS-SM9-RIBS' s update time remains under 15 seconds, while BF-SM9-RIBS takes about 45 seconds when $N = 8192$.

Comparison with International Schemes: Table 3 compares CS-SM9-RIBS with other revocable IBS schemes [?, ?]. Except [?] (based on trilinear maps), others use standard bilinear maps. CS-SM9-RIBS has smaller temporary keys and signatures than most schemes. Its signing time requires one more exponentiation than [?], but [?]' s update complexity grows linearly with N. [?] depends only on R but relies on less mature trilinear maps. Overall, CS-SM9-RIBS demonstrates clear performance advantages.

Table 3: Comparison with Other Revocable IBS Algorithms

Scheme	Temp Key Size	Verification Time	Update Key Time
[?]	$3\|\|p\|\|$	$3P + 3E$	$O(N - R)$E
[?]	$2\|\|p\|\|$	$2P + 4E$	$(\log(N/R))$E
[?]	$2\|\|p\|\|$	$1P + 3E$	$O(N - R)$E
[?]	$2\|\|p\|\|$	$3P$	$O(R)$E
CS-SM9-RIBS	$2\|\|p\|\|$	$2P + 4E$	$(\log(N/R))$E

5 Conclusion

This paper proposes a revocable SM9 identity-based signature algorithm addressing SM9' s key revocation problem. Using a complete subtree, it achieves efficient user privilege control and key revocation/update. Theoretical and experimental analyses show CS-SM9-RIBS outperforms BF-SM9-RIBS in key update efficiency. Future work includes reducing the doubled signing/verification time overhead.

References:

[1] Shamir A. Identity-based cryptosystems and signature schemes [C]// Proc of the 4th Advances in Cryptology-Crypto. Berlin: Springer, 1984: 47-53.

[2] Sakai R, Ohgishi K, Kasahara M. Cryptosystems based on pairing [J]. Symposium Cryptographic Information Security, 2000 (43): 26–28.

[3] Boneh D, Franklin M. Identity-based encryption from the Weil pairing [C]// Proc of the 21th Advances in Cryptology-Crypto. Berlin: Springer, 2001: 213-229.

[4] Cocks C. An identity-based encryption scheme based on quadratic residues [C]// Proc of the IMA international conference on cryptography and coding. Berlin: Springer, 2001: 360-363.

[5] Hofheinz D, Jia D, Pan J. Identity-based encryption tightly secure under chosen-ciphertext attacks [C]// Proc of the 29th Advances in Cryptology–Asiacrypt. Berlin: Springer, 2018: 190-220.

[6] Langrehr R, Pan J. Tightly secure hierarchical identity-based Encryption [J]. Journal of Cryptology, 2020, 33 (4): 1787-1821.

[7] Sahai A, Waters B. Fuzzy identity-based encryption [C]// Proc of the 11th Advances in Cryptology–Eurocrypt. Berlin: Springer, 2005: 457-473.

[8] Waters B. Efficient identity-based encryption without random oracles [C]// Proc of the 11th Advances in Cryptology–Eurocrypt. Berlin: Springer, 2005: 114-127.

[9] Cheng Zhaohui. The SM9 cryptographic schemes [J]. Cryptology ePrint Archive, 2017.

[10] ISO/IEC 14888-3: 2018, Information technology security techniques: digital signatures with appendix Part 3: Discrete logarithm based mechanisms [S]. 2018.

[11] ISO/IEC 18033-5: 2015/AMD 1: 2021 Information technology security techniques: encryption algorithms Part 5: Identity-based ciphers Amendment 1: SM9 mechanism [S]. 2021.

[12] Ji Honghan, Zhang Hongjie, Shao Lisong, et al. An efficient attribute-based encryption scheme based on SM9 encryption algorithm for dispatching and control cloud [J]. Connection Science, 2021, 33 (4): 343-354.

[13] Mu Yongheng, Xyu Haixia, Li Peili, et al. Secure two-party SM9 signing [J]. Science China Information Sciences, 2020, 63 (8): 1-3.

[14] Zhang Yyu, Sun Guangmin, Li Yyu. Research on Mobile Internet Authentication Scheme Based on SM9 Algorithm [J]. Netinfo Security, 2021, 21 (4): 1-9.

[15] Yao Yingying, Chang Xiaolin, Zhen Ping. Decentralized identity authentication and key management scheme based on blockchain [J]. Cyberspace Security, 2019, 10 (6): 33-39.

[16] MA Xiaoting, MA Wenping, LIU Xiaoxue. A cross domain authentication scheme based on blockchain technology [J]. Acta Electonica Sinica, 2018, 46 (11): 2571.

[17] Boldyreva A, Goyal V, Kumar V. Identity-based encryption with efficient revocation [C]// Proc of the 15th ACM conference on Computer and communications security, New York: ACM Press, 2008: 417-426.

[18] Libert B, Vergnaud D. Adaptive-ID secure revocable identity-based encryption [C]// Proc of the 9th Cryptographers' Track at the RSA Conference. Berlin: Springer, 2009: 1-15.

[19] Seo J H, Emura K. Revocable identity-based cryptosystem revisited: security models and constructions [J]. IEEE Trans on Information Forensics and Security, 2014, 9 (7): 1193-1205.

[20] Lee K, Lee D H, Park J W. Efficient revocable identity-based encryption via subset difference methods [J]. Designs, Codes and Cryptography, 2017, 85 (1): 39-76.

[21] Katsumata S, Matsuda T, Takayasu A. Lattice-based revocable (hierarchical) IBE with decryption key exposure resistance [J]. Theoretical Computer Science, 2020, 809: 103-136.

[22] Takayasu A, Watanabe Y. Lattice-based revocable identity-based encryption with bounded decryption key exposure resistance [C]// Proc of Australasian Conference on Information Security and Privacy. Berlin: Springer, 2017: 184-204.

[23] Li Jin, Li Jingwei, Chen Xiaofeng, et al. Identity-based encryption with outsourced revocation in cloud computing [J]. IEEE Trans on Computers, 2013, 64 (2): 425-437.

[24] Qin Baodong, Deng R H, Li Yingjiu, et al. Server-aided revocable identity-based encryption [C]// European Symposium on Research in Computer Security. Berlin: Springer 2015: 286-304.

[25] Qin Baodong, Liu Ximeng, Wei Zhuo, et al. Space efficient revocable IBE for mobile devices in cloud computing [J]. Science China Information Sciences, 2020, 63 (3): 1.

[26] Ma Xuecheng, Lin Dongdai. Generic constructions of revocable identity-based encryption [C]// Proc of the International Conference on Information Security and Cryptology. Berlin: Springer, 2019: 381-396.

[27] Lai Jianchang, Huang Xinyi, He Debiao, et al. Security analysis of SM9 digital signature and key encapsulation [J]. SCIENTIA SINICA Informationis, 2021, 51 (11): 1900-1913.

[28] Wei Jianghong, Liu Wenfen, Hu Xuexia. Forward-secure identity-based signature with efficient revocation [J]. International Journal of Computer Mathematics, 2017, 94 (7): 1390-1411.

[29] Yang Xiaodong, Ma Tingchun, Yang Ping, et al. Security analysis of a revocable and strongly unforgeable identity-based signature scheme [J]. Information Technology and Control, 2018, 47 (3): 575-587.

[30] Zhao Jing, Wei Bin, Su Yang. Communication-efficient revocable identity-based signature from multilinear maps [J]. Journal of Ambient Intelligence and Humanized Computing, 2019, 10 (1): 187-198.